Computergrafik 2: Übung 6

Korrelation im Orts- und Frequenzraum, Filtern im Frequenzraum, Wiener Filter

Quiz

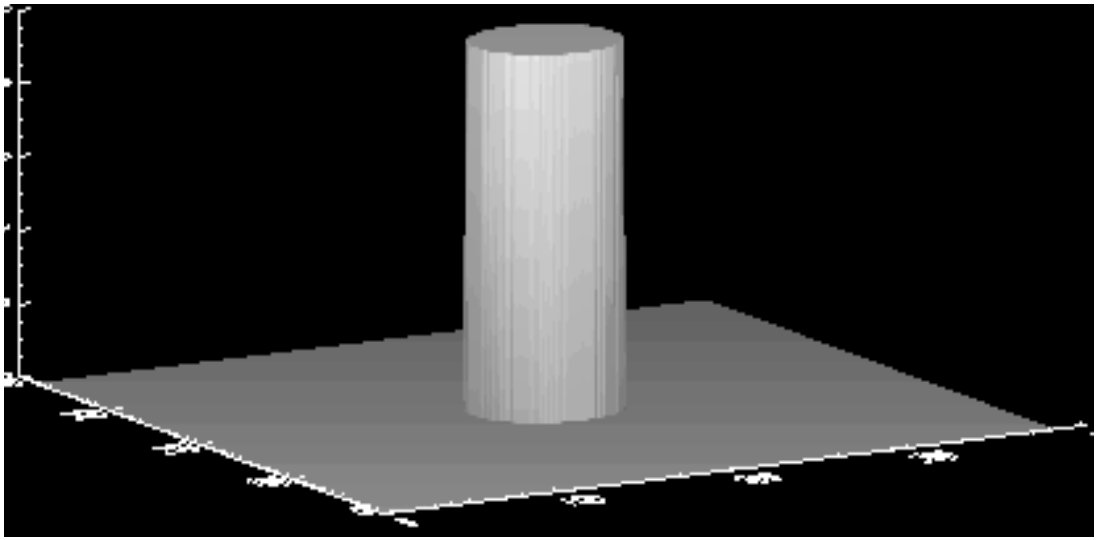
- Warum Filtern im Frequenzraum?
- Ideales Tiefpassfilter? Parameter? Eigenschaften?
- Butterworth-Filter? Parameter? Eigenschaften?
- Gaussfilter? Parameter? Eigenschaften?
- Grundidee Bandpass- / Bandreject-Filterung?
- Grundidee inverse Filterung?
- Probleme bei der inversen Filterung?

Besprechung Übung 5

- Probleme?

Ideales Tiefpassfilter

- Tiefpassfilter lässt tiefe Frequenzen passieren und dämpft hohe Frequenzen
- Ideales Tiefpassfilter $H_{F_{\max}}(u,v) = \begin{cases} 1 & \text{, falls } u^2 + v^2 \le F_{\max}^2 \\ 0 & \text{, sonst.} \end{cases}$
  - ideal low pass filter (ILPF)
- F_{max}: Cut-Off-Frequenz

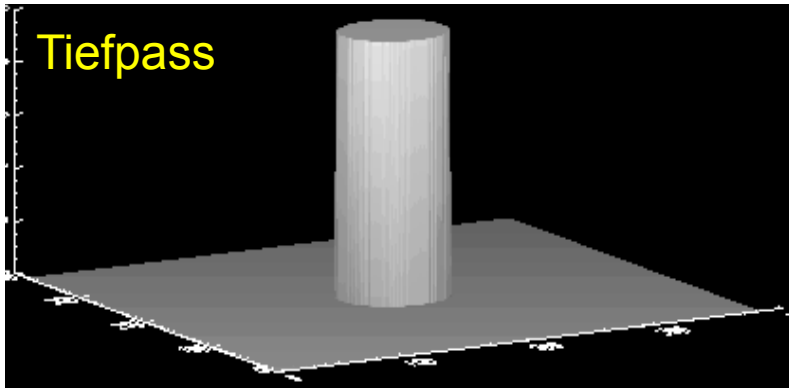


© K. D. Tönnies, Grundlagen der Bildverarbeitung

Butterworth-Filter

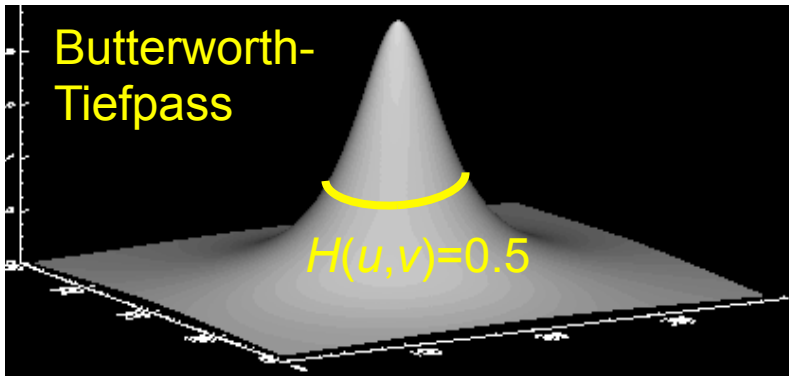
- Frequenzen werden nicht gelöscht, sondern nur abgeschwächt
- Tiefpass-Filter (BLPF):

$$H(u,v) = \frac{1}{1 + \left(D(u,v)/D_0\right)^{2n}}$$



- Hochpass-Filter (BHPF):

$$H(u,v) = \frac{1}{1 + \left(D_0/D(u,v)\right)^{2n}}$$



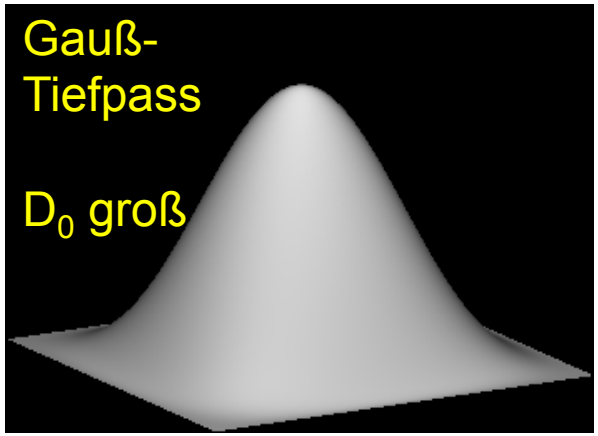
D_0: Cutoff-Frequenz

$D(u,v)$: Frequenz, Abst. Ursprung

n: Ordnung des Filters

Gauß-Filter

- Keine Artefakte, da Fourier-Transformation einer Gauß-Funktion wieder eine Gauß-Funktion

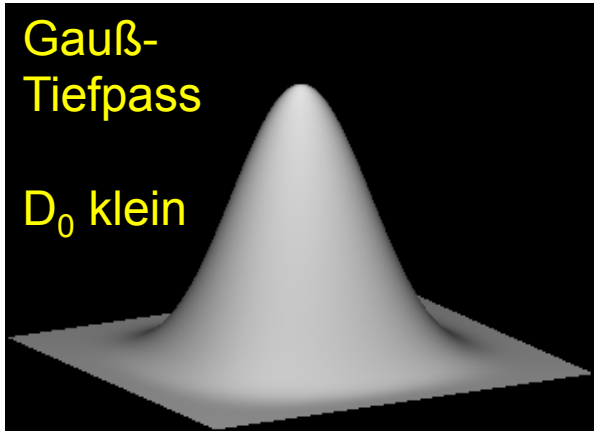


- Tiefpass-Filter (GLPF):

$$H(u,v) = \exp\left(-\frac{D^2(u,v)}{2D_0^2}\right)$$

- Hochpass-Filter (GHPF):

$$H(u,v) = 1 - \exp\left(-\frac{D^2(u,v)}{2D_0^2}\right)$$



D_0: entspricht σ

$D(u,v)$: Frequenz, Abstand vom Ursprung

Vermeiden der Auslöschung niedriger Frequenzen beim GHPF

- Hochpass-Filter (GHPF):

$$H(u,v) = 1 - \exp\left(-\frac{D^2(u,v)}{2D_0^2}\right)$$

$0 \le H(u,v) \le 1$

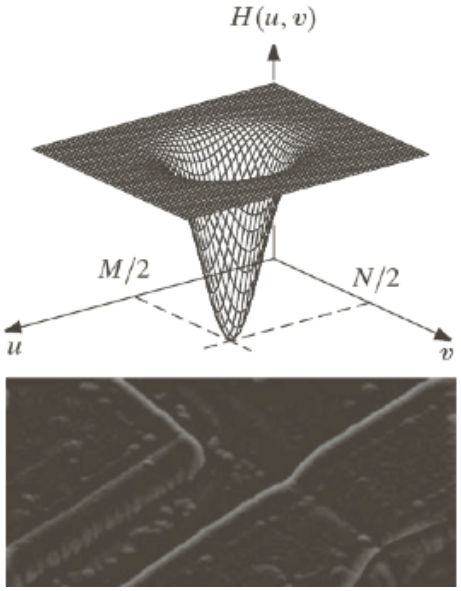
- modifiziertes GHPF:

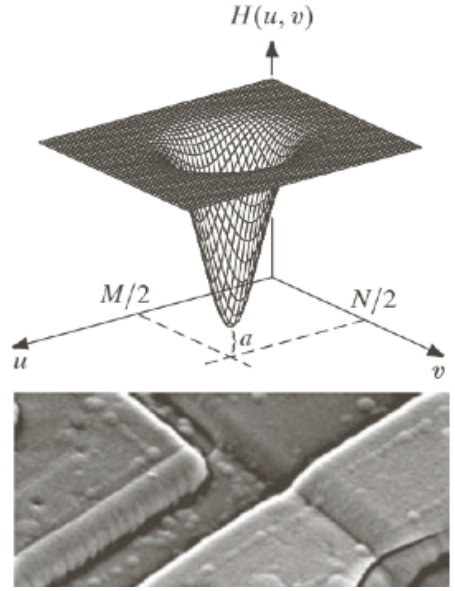
$$H_a(u,v) = a + (1-a)\cdot H(u,v)$$

$\mathbf{a} \le H_a(u,v) \le 1$

D_0: entspricht σ

$D(u,v)$: Frequenz, Abstand vom Ursprung



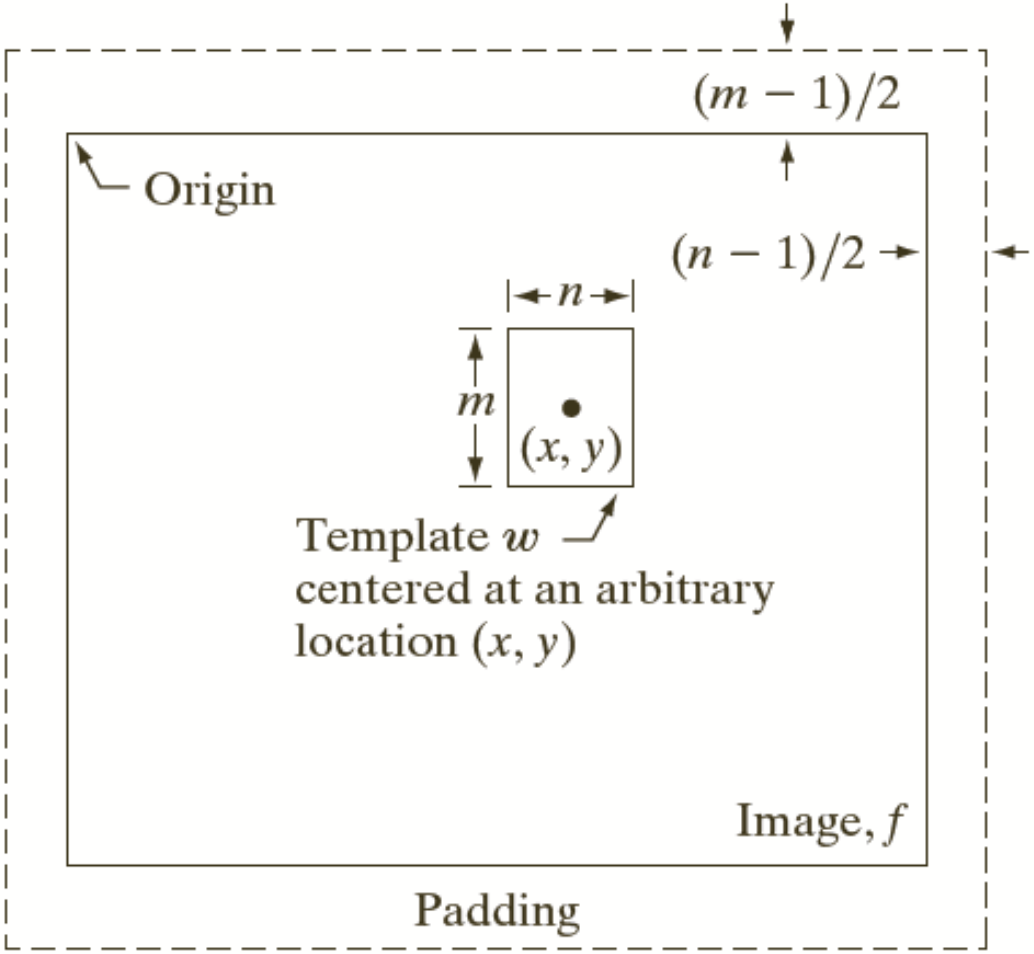


in Python

```
def GLPF((h,w), D0, a = 0.0):
    H = np.zeros((h,w), dtype='float32')
    for y in xrange(h):
        for x in xrange(w):
            dx = x - w/2
            dy = y - h/2
            D = np.sqrt(dx * dx + dy * dy)
            H[y,x] = np.exp(-D**2/(2*D0**2))
            if a > 0.0:
                H[y,x] = a + (1.0 - a) * H[y,x]
    return H
```

Korrelation im Ortsraum

- Ähnlichkeiten zwischen Bild und Modell feststellen
- Modell (Template) im Bild suchen



© R. C. Gonzalez & R. E. Woods, Digital Image Processing

Korrelation im Ortsraum

- Ähnlichkeitsmaß (normalisierter) Korrelationskoeffizient
  - Mittelwerte subtrahiert und Varianzen normiert
- Kleineres Bild pixelweise über größeres Bild verschieben und Korrelationskoeffizient berechnen

  (f = Bild, m = Modell/Template, (x,y) = Suchposition)

$$cc_{f,m}(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(m(s,t)-\overline{m}\right)\left(f(x+s,y+t)-\overline{f}(x,y)\right)}{\sqrt{\left[\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(m(s,t)-\overline{m}\right)^2\right]\left[\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(f(x+s,y+t)-\overline{f}(x,y)\right)^2\right]}}$$

$$-1 \le cc_{f,m}(x,y) \le 1$$

Korrelation im Frequenzraum

- Subtrahiere Mittelwerte von f und m
- Ähnlichkeitsmaß: Korrelationskoeffizient $cc_{f,g}$

$$cc_{f,m}(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(m(s,t)-\overline{m}\right)\left(f(x+s,y+t)-\overline{f}(x,y)\right)}{\sqrt{\left[\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(m(s,t)-\overline{m}\right)^2\right]\left[\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(f(x+s,y+t)-\overline{f}(x,y)\right)^2\right]}}$$

$$= k\sum_{s=-a}^{a}\sum_{t=-b}^{b}\left(m(s,t)-\overline{m}\right)\left(f(x+s,y+t)-\overline{f}(x,y)\right)$$

$$= k\sum_{s=-a}^{a}\sum_{t=-b}^{b} m(s,t)f(x+s,y+t) \Longrightarrow FT\left(\left[f\circ g\right](x,y)\right) = F(u,v)\cdot G^*(u,v)$$

Korrelationsfunktion — Korrelation — G konjugiert

Python-Tipps

```
plt.gray()
…
f = f - np.mean(f)
fftSize = (M,N)
F = np.fft.fft2(f, fftSize)
F = np.fft.fftshift(F)
plt.imshow(np.log(np.abs(F)))
…
M = np.conjugate(M)
…
```

```
CC = np.fft.ifftshift(CC)
cc = np.fft.ifft2(CC, fftSize)
cc = cc[0:h, 0:w]
cc = np.real(cc)
…
plt.imshow(cc,
cmap=plt.get_cmap('jet'))
…
cc2[cc2 < 0.75 * np.max(cc2)] = 0.0
```

Bandreject/Bandpass-Filter

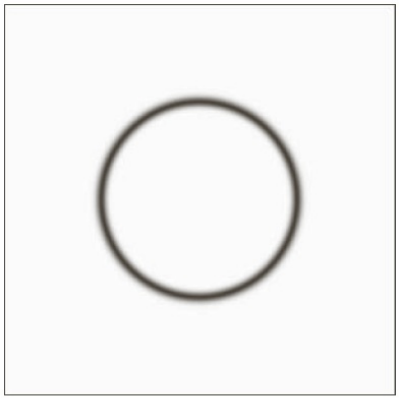
- Bandreject-Filter
  - Ideal
  - Butterworth
  - Gauß

$$H(u,v) = \begin{cases} 0 & \text{, falls } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{, sonst} \end{cases}$$

$$H(u,v) = \frac{1}{1 + \left(\frac{DW}{D^2 - D_0^2} \right)^{2n}}$$

$$H(u,v) = 1 - \exp\left(-\left(\frac{D^2 - D_0^2}{DW} \right)^2 \right)$$

Gauß-Bandreject-Filter



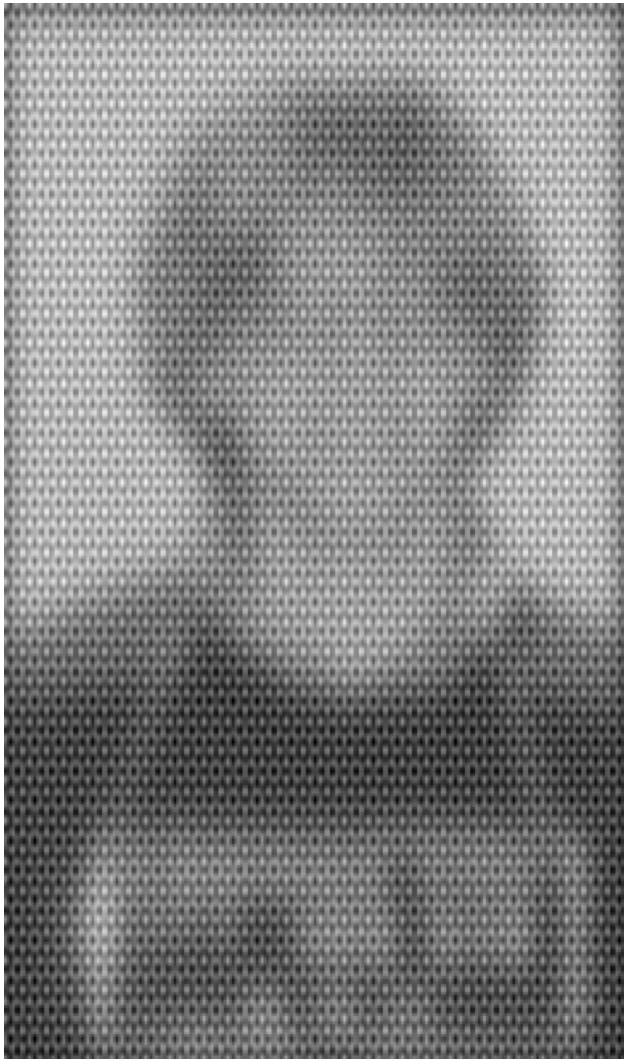
Gauß-Bandpass-Filter

- Bandpass-Filter

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

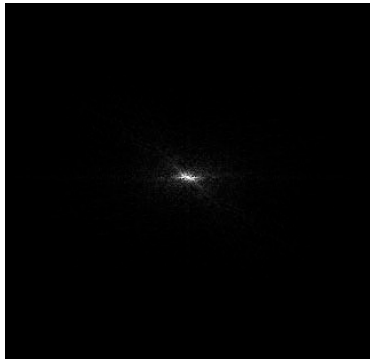
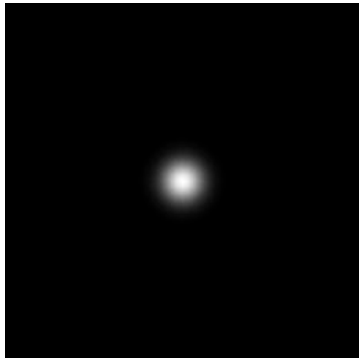
Wer ist das?

- Bildrestauration, wie?



Inverse Filterung

- Vollständige Rückgewinnung der Information aus den gestörten Daten

$FT^{-1}($  $/$  $)=$ 

Numerische Probleme bei der inversen Filterung

$$g = h * f \Rightarrow f(m,n) = FT^{-1}\left(\frac{G(u,v)}{H(u,v)}\right)$$

komplexe Zahlen dividieren?

- Problem: Nullstellen von H
  - Treten auf, falls h als Matrix nicht den vollen Rang hat
  - Auch kleine Werte von H sind numerisch schon ein Problem
- Deswegen in der Praxis:

$$F(u,v) = \begin{cases} \frac{G(u,v)}{H(u,v)} & H(u,v) > H_{\min} \\ 0 & sonst \end{cases}$$

Rauschen

- Problem: Inverse Filterung geht von idealen (ungestörten) Daten aus
- aber: Bilddaten enthalten Rauschen
- inverse Filterung verstärkt Rauschen extrem
  - mit steigender Frequenz: (weißes) Rauschen bleibt, Signal-Amplitude nimmt schnell ab, Rauschanteil wird höher

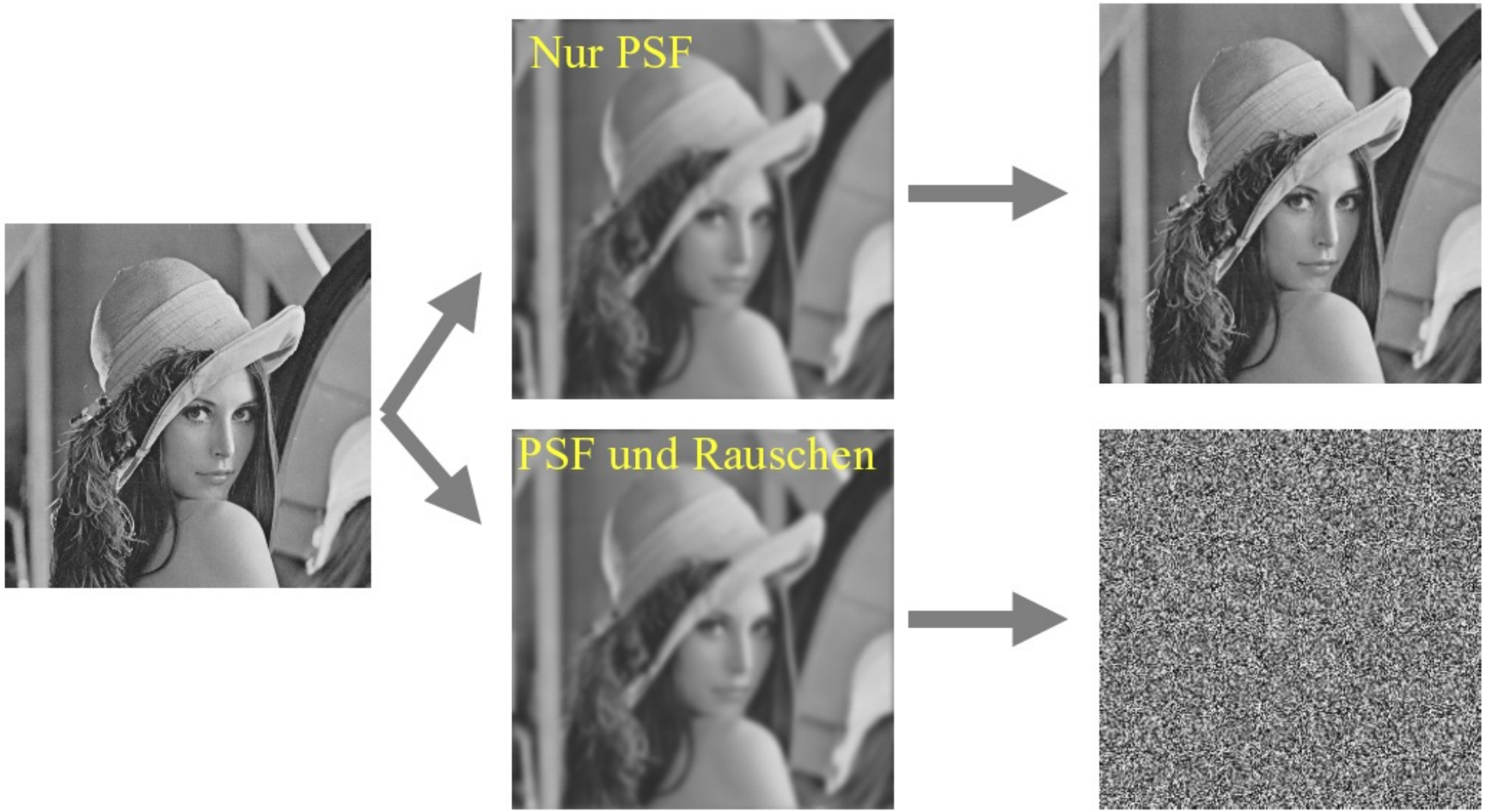
$$g(m,n) = f(m,n) * h(m,n) + \eta(m,n)$$

$$G(u,v) = F(u,v) \cdot H(u,v) + N(u,v) \Leftrightarrow \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

  - ad-hoc Lösung: hohe Frequenzen ausschließen

Rauschen

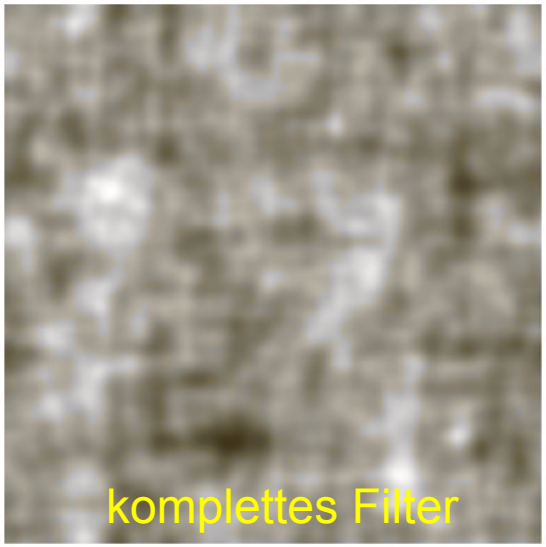
- Invertierung bei Rauschen oft nicht möglich

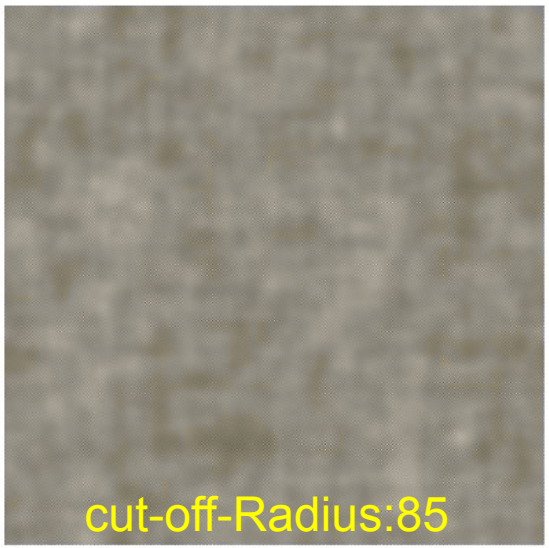


Abschneiden hoher Frequenzen

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.





Wiener Filter

- Minimierung des Fehlers zwischen Originalbild f und Schätzer $\hat{f}$ führt zu

$$\hat{F}(u,v) = X(u,v) \cdot G(u,v)$$

$$X(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_\eta(u,v)}{S_f(u,v)}} = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}}$$

- S_η und S_f sind die Spektren (Quadrate der Amplituden) des Rauschens bzw. der ungestörten Funktion
  - $S_\eta = 0$ (ungestört) ➔ perfekte inverse Filterung
- Wiener Filter dämpft Frequenzen abhängig von SNR

Heuristisches Wiener Filter

- Leider ist S_η in der Praxis meist unbekannt
- Lösung: Konstante K: heuristisches Wiener Filter

$$\hat{F}_K(u,v) = X_K(u,v) \cdot G(u,v)$$

$$X_K(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}$$

$$|H(u,v)|^2 = H^*(u,v)H(u,v) = re(H(u,v))^2 + im(H(u,v))^2$$

Wiener Filter Herleitung

- Bild/Signal s und Rauschen/Noise n im Ortsraum:

$$o(x) = s(x) + n(x)$$

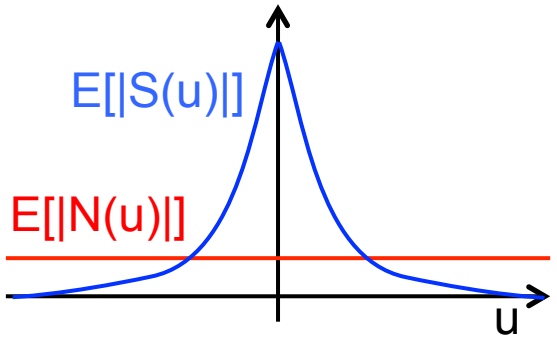
o = „Observation“

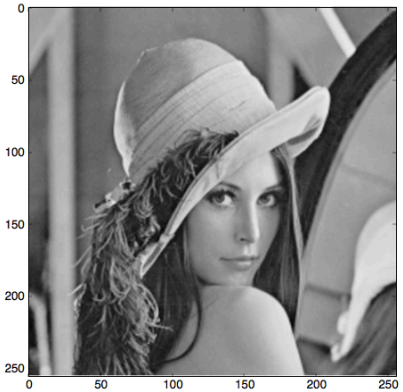
- Bild/Signal s und Rauschen/Noise n im Frequenzraum:

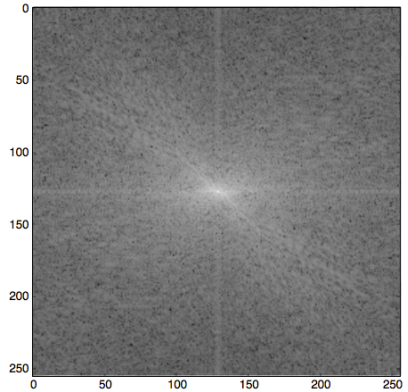
$$O(u) = S(u) + N(u)$$

- Annahmen
  - S(u) sind normalverteilte Zufallsvariablen
  - N(u) sind normalverteilte Zufallsvariablen
  - E[|S(u)|] nimmt mit steigendem u ab
  - E[|N(u)|] ist konstant (weißes Rauschen)

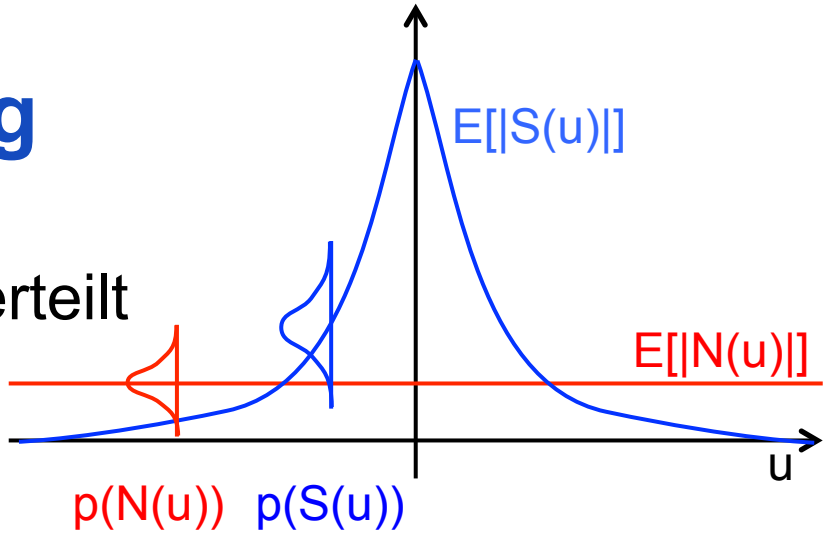
E[X] = Erwartungswert der Zufallsvariablen X







Wiener Filter Herleitung



- S(u) ist $(\mu_S(u),\sigma_S^2(u))$-normalverteilt

$$p\left(S(u)\right)=\frac{1}{\sigma_S(u)\sqrt{2\pi}}\exp\left(-\frac{\left(S(u)-\mu_S(u)\right)^2}{2\sigma_S(u)^2}\right)$$

$$\sigma_S(u)^2=E\left[\left|S(u)-\mu_S(u)\right|^2\right]=E\left[\left|S(u)\right|^2\right],\ \textit{für}\ \mu_S(u)=0$$

- N(u) ist $(\mu_N(u),\sigma_N^2(u))$-normalverteilt

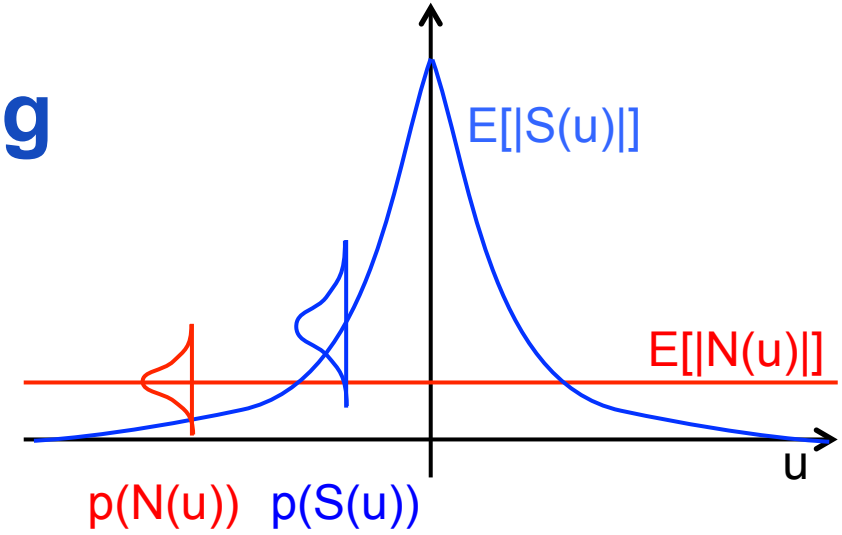
$$\textit{für}\ \mu_N(u)=0,\sigma_N(u)=const$$

$$p\left(N(u)\right)=\frac{1}{\sigma_N(u)\sqrt{2\pi}}\exp\left(-\frac{\left(N(u)-\mu_N(u)\right)^2}{2\sigma_N(u)^2}\right)=\frac{1}{\sigma_N\sqrt{2\pi}}\exp\left(-\frac{N(u)^2}{2\sigma_N^2}\right)$$

$$O(u)=S(u)+N(u)$$

$$p\left(O(u)\,|\,S(u)\right)=p\left(S(u)+N(u)\,|\,S(u)\right)=\frac{1}{\sigma_N\sqrt{2\pi}}\exp\left(-\frac{\left(O(u)-S(u)\right)^2}{2\sigma_N^2}\right)$$

Wiener Filter Herleitung



- Satz von Bayes

$$p\big(S(u)\,|\,O(u)\big) = \frac{p\big(O(u)\,|\,S(u)\big)\cdot p\big(S(u)\big)}{p\big(O(u)\big)}$$

$$p\big(O(u)\big) = \sum_{v=-\infty}^{\infty} p\big(O(u)\,|\,S(u)=v\big)\cdot p\big(S(u)=v\big) = const$$

- Gegeben Beobachtung O(u), was ist wahrscheinlichstes Signal?
  - maximiere p(S(u)|O(u))
  - äquivalent zu: minimiere –log(p(S(u)|O(u)))

$$-\log\Big(p\big(S(u)\,|\,O(u)\big)\Big) = -\log\Big(p\big(O(u)\,|\,S(u)\big)\Big) - \log\Big(p\big(S(u)\big)\Big) + \log\Big(p\big(O(u)\big)\Big)$$

$$= \frac{\big(O(u)-S(u)\big)^2}{2\sigma_N^2} + \frac{\big(S(u)-\mu_S(u)\big)^2}{2\sigma_S(u)^2} + const$$

Abl. nach S(u)

$$S(u) = \frac{1}{1+\dfrac{\sigma_N^2}{\sigma_S(u)^2}}\,O(u)$$

Wiener Filter Herleitung

$$f(S(u)) = -\log\left(p\left(S(u)\,|\,O(u)\right)\right)$$

$$= -\log\left(p\left(O(u)\,|\,S(u)\right)\right) - \log\left(p\left(S(u)\right)\right) + \log\left(p\left(O(u)\right)\right)$$

$$= -\log\left(\frac{1}{\sigma_N\sqrt{2\pi}}\exp\left(-\frac{\left(O(u)-S(u)\right)^2}{2\sigma_N^2}\right)\right) - \log\left(\frac{1}{\sigma_S(u)\sqrt{2\pi}}\exp\left(-\frac{\left(S(u)\right)^2}{2\sigma_S(u)^2}\right)\right) + \log\left(p\left(O(u)\right)\right)$$

$$= -\log\left(\frac{1}{\sigma_N\sqrt{2\pi}}\right) + \frac{\left(O(u)-S(u)\right)^2}{2\sigma_N^2} - \log\left(\frac{1}{\sigma_S(u)\sqrt{2\pi}}\right) + \frac{\left(S(u)\right)^2}{2\sigma_S(u)^2} + \log\left(p\left(O(u)\right)\right)$$

- Ableitung von f(S(u)) = –log(S(u)|O(u)) nach S(u):

$$f'\left(S(u)\right) = -2\frac{O(u)-S(u)}{2\sigma_N^2} + 2\frac{S(u)}{2\sigma_S(u)^2} = S(u)\left(\frac{1}{\sigma_N^2} + \frac{1}{\sigma_S(u)^2}\right) - \frac{O(u)}{\sigma_N^2} = 0$$

$$\Leftrightarrow S(u) = \frac{1}{1+\frac{\sigma_N^2}{\sigma_S(u)^2}}O(u) = W(u)\cdot O(u), \quad mit\ W(u) = \frac{1}{1+\frac{\sigma_N^2}{E\left[\left|S(u)\right|^2\right]}}$$